Management of Ubiquitous Systems with a Mobile Application Using Discrete Event Simulations

Souhila Sehili
University of Corsica
SPE UMR CNRS 6134
Campus Grimaldi, 20250
Corte (France)
sehili@univ-corse.fr

Laurent Capocchi
University of Corsica
SPE UMR CNRS 6134
Campus Grimaldi, 20250
Corte (France)
capocchi@univ-corse.fr

Jean-François Santucci
University of Corsica
SPE UMR CNRS 6134
Campus Grimaldi, 20250
Corte (France)
santucci@univ-corse.fr

ABSTRACT

Discrete-event simulation plays an increasingly important role in the management of ubiquitous systems. This working progress proposes a cross-platform mobile application dedicated to the management of ubiquitous systems using discrete-event simulation. The mobile application aims to perform the simulation of models specified with discrete-event system specifications (DEVS). Web services are invoked by the mobile application in a generic way in order to load, set and simulate models defined using the DEVSimPy software and stored on a web server. The propose approach has been validated on an ubiquitous system defined from an interconnection of physical computing platforms including microcontrollers, switches and sensors.

Categories and Subject Descriptors

I.6.8 [**Simulation and Modeling**]: Types of Simulation—*Discrete event*; D.2.6 [**Software Engineering**]: Programming Environments—*Graphical environments*

General Terms

Theory

Keywords

Discrete event modeling, Simulation, Ubiquitous system, DEVSimPy, Mobile application

1. INTRODUCTION

Ubiquitous systems [4] necessary lean on an interaction between virtual objects and users. Discrete-event simulation can offer a way to control such systems according to real time constraints [5, 3]. The paper deals with a mobile application aimed to manage discrete-event simulations obtained from DEVS (Discrete EVent system Specification) models associated with connected objects such as board computers, sensors, controllers or actuators.

SIGSIM-PADS'15, June 10–12, 2015, London, United Kingdom.
ACM 978-1-4503-3583-6/15/06.
http://dx.doi.org/10.1145/2769458.2769490.

The interest of the propose working progress is to strongly associate simulations, mobile applications and connected objects. The result will be the ability to manage connected objects (sensors, computer boards, actuators, controller) from a mobile application while providing intelligent decisions based on simulations. The user of the proposed mobile application DEVSimPy-Mob will first have to connect to a board computer through the cloud. This connection will offer the user a list of DEVS models that can be simulated. These models are based on the DEVS formalism and involved a set of DEVS models in order to manage sensors, actuators, board computers or controller. One of the main interests of such an approach is the possibility to associate DEVS models of connected objects with classical DEVS models (such as prediction models, decision models, etc.) allowing to propose the management of connected objects from a mobile application integrating intelligent decisions based on simulations.

The DEVS modeling aspects are implemented using the DEVSimPy Framework [1]. The connected objects features are based on Phidgets [2]. Phidgets offer a set of low-cost electronic components and sensors that are controlled by a personal computer. The integration of Phidgets interface with DEVS is necessary for two reasons: (i) to simulate real time systems with data coming from Phidgets sensors; (ii) to exploit simulation results to control in real time Phidgets actuators; (iii) to enrich the use of Phidgets interfaces by coupling them with DEVS models specialized in decision making (neural network, fuzzy logic, optimization via simulation, etc.). The rest of the paper is organized as follows: the next section deals with the Modeling and Simulation methodologies that have been developed in the framework of the presented work. The implementation and results are presented in section 3 while the the future work and conclusion sare detailed in section 4 and 5.

2. M&S METHODOLOGIES

DEVS (Discrete EVent system Specification) [6] has been introduced as an abstract formalism for the modeling of discrete event systems, and allows a complete independence from the simulator using the notion of abstract simulator. DEVS defines two kinds of models: atomic models and coupled models. An atomic model is a basic model with specifications for the dynamics of the model. It describes the behavior of a component, which is indivisible, in a timed state transition level. Coupled models tell how to couple several component models together to form a new model. This kind of model can be employed as a component in a larger coupled model. As in general systems theory, a DEVS model contains a set of states and transition functions that are triggered by the simulator. DEVSimPy [1] is an open Source project supported by the SPE team of the university of Corsica. This aim is to provide a GUI for the modeling and simulation of PyDEVS models. PyDEVS is an Application Programming Interface (API) allowing the implementation of the DEVS formalism in Python language.

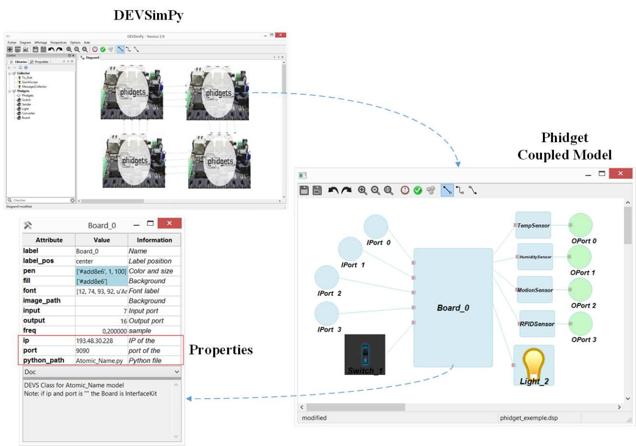

Figure 1: The DEVSimPy Phidget Library.

From a M&S aspects the following tasks have been performed: (i) Development of a DEVSimPy Phidget component library for manipulating Phidget board computers, sensors and actuators. The Figure 1 illustrates the definition and use of the Library (with components "Board", a To-disk corresponding to a "Lamp" component and a "RandomGenerator" component corresponding to a "Switch" component). Figure 1 points out how a Phidget DEVSimPy model is automatically associated with sensors, actuators and boards through the IP address of the Phidget. (ii) Implementation of a Web Server in order to run the simulations involving components of the DEVSimPy Phidgets library which allows to manage connected objects and provision of web services allowing to dynamically interact from the mobile application with the DEVSimPy framework (from the Web Server).

3. IMPLEMENTATION AND RESULTS

The implementation has involved two parts: (i) the development of the mobile application and (ii) the M&S features. The chosen technologies concerning the development of the mobile application are summarized as follows: (i) Technology choice: PhoneGap, JavaScript for the communication with the DEVSimPy WebServices; (ii) Tools: Bookstore, JointJS, CSS: ratchet framework. The technical aspects of the implementation DEVSimPy library and DEVSimPy Web Server are listed below: (i) Development of nogui DEVSimPy version; (ii) Implementation of a python Web Server: Python, HTML5, CSS3, MySQL; (iii) Servers: Web Server, File server and SVN server. The capabilities of the mobile application DEVSimPy-Mob are summarized below: (i) Multi-Platform (IOS, Android, etc.); (ii) Access to the Web Server Services DEVSimPy; (iii) Launch simulations (interactions with the simulation); (iv) Graphically visualization of DEVS models; (v) Visualization of the results (as sensors data, the activity of actuators as the state of a lamp managed by the mobile application).

4. FUTURE WORK

During a short term period, we plan to improve the visualization of the results and to develop a set of web-services to better communicate with the DEVSimPy web server. In a longer term research, we plan to deal with complex applications involving the interconnections connected objects such as sensors, actuators, board computers and DEVS models specialized with artificial intelligent features such as neural networks, fuzzy inductive modeling, etc.

5. CONCLUSIONS

The proposed work in progress concerns the development of an approach for managing ubiquitous systems from a mobile application using DEVS. We have proposed a library of DEVSimPy components allowing to model and simulate the behavior of connected objects. Since the DEVSimPy models are associated with the real objects through the IP address, we are able to manage connected objects through DEVS simulations. Furthermore DEVS models are accessible through the Internet using a DEVSimPy Web server. We have also developed a generic mobile application allowing to: (i) select a DEVS model to be simulated, (ii) performs the simulation and (iii) visualizes the obtained results.

6. REFERENCES

[1] Capocchi, L., Santucci, J.-F., Poggi, B. and Nicolai, C. 2011. DEVSimPy: A Collaborative Python Software for Modeling and Simulation of DEVS Systems, In *Proc. of 20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (Paris, France, 27-29 June, 2011). IEEE, 170-175. DOI=10.1109/WETICE.2011.31.

[2] Greenberg, S. and Fitchett, C. 2001. Phidgets: Easy Development of Physical Interfaces Through Physical Widgets, In *Proc. of the 14th Annual ACM Symposium on User Interface Software and Technology* (Orlando, Florida, 2001). ACM, New York, NY, USA, 209-218. DOI=10.1145/502348.502388.

[3] Jeffery, A., Panke, J., Jonathon, E., Eaket, N. and Wainer, G. 2013. Mobile Simulation with Applications for Serious Gaming, In *Proc. of the Summer Computer Simulation Conference* (Toronto, Ontario, Canada 2013). Society for Modeling & Simulation International, Vista, CA, Art. 27, 8 pages. ISBN=978-1-62748-276-9.

[4] Kindberg, T. and Fox, A. 2002. System software for ubiquitous computing. *IEEE Pervasive Computing 1*, 1 (January 2002), 70-81. DOI=10.1109/MPRV.2002.993146.

[5] Sarjoughian, H. S., Gholami, S., Jonathon, E. and Jackson, T. 2013. Interacting Real-time Simulation Models and Reactive Computational-physical Systems, In *Proc. of the Winter Simulation Conference: Simulation: Making Decisions in a Complex World* (Washington, D.C., 8-11 Dec, 2013). IEEE, 1120-1131. DOI=10.1109/WSC.2013.6721501.

[6] Zeigler, B., Praehofer, H. and Kim, T.G. 2000. *Theory of Modeling and Simulation, Second Edition.* Academic Press, Inc., Orlando, FL, USA.